The Use of Precise Specifications in the Development of Software

Bill Wilson
Justin McBride
CSE870, Spring 2002

Parnas' Objective

- Show the shortcomings of the current (1977) "foundational research."
- Dispute the connotations of commonly misused terms: descriptions, specifications, models, prototypes.
- Demonstrate how simple mathematics can be used to describe programs.
- Create a foundation for future work.

Overview for Parnas Paper

- Two versions of the same paper
  - IFIP- 1977
Attack on Current Work in Formal Methods
(Overview Cont.)
- Refers to it as “attractive.”
  - Not fruitful.
    - Too complicated for common program.
    - Need consistent mathematical formulations to describe programs.

Terms Commonly Misused
(Overview Cont.)
- Description
- Specification
- Model
- Prototype

Four Types of Programs
(Overview Cont.)
- Real-Time
  - Non-Terminating programs
  - Infinite sequence of inputs
- Terminating
  - Often used as components of larger programs
Four Types of Programs
(Overview Cont.)
- Modules
  - Collection of programs that provide information hiding for the data structures
- Objects
  - Used to produce copies of data structures

Technique Overview
(Overview Cont.)
- Information Hiding Modules
- Limited Domain Relationships
- Definition of Program Domain Language
- Traces for Black Box Descriptions

Impact
- Created a foundation for work in formal specifications.
- Help to grow and expand the formal software engineering community.
- Became the basis for numerous future works.
Formal Specification: Roadmap
Axel van Lamsweerde
- Show the short comings in all of the current (now 2000) techniques in formal specification development
- How does it extend Parnas' work?

Specifying Software Requirements...
Kathryn Heninger
- Demonstrate that proper SE techniques improve application usage.
- Demonstrate that proper SE techniques do no have to interfere with performance.
- How does it extend Parnas' work?

Tam '97
Iglewski, Kubica, Madey, Mincer-Daszkiewicz, Stencel
- Provides SE reference manual for the Trace Assertion Method.
- Provide an overview of how traces work.
- Make Trace Assertion Method an effective tool.
- How does it extend Parnas' work?
Formal Specification of Software

Greg Jones.
High-Level overview of formal specification

Formal Specification of Software (Technique Overview)

Differentiate between informal and formal Software Specification.
Give Practical examples of formal specifications.
Discuss why formal specifications are not commonly used.
Concurrently generating proofs of correctness.

Why It Should Have Cited Parnas' Work

Contain three main points that Parnas had made in 1977.
- Proof of correctness
- Agrees with points on which other authors cite Parnas' work
- Behavioral vs. Constructive Descriptions